

Ing. Fabián Talio  
Ing. Carolina Zapata

Lic. Jorge Pérez Herrera  
Ing. Matías Varela

Lic. Carina Povarchik  
Prof. Lucas Candia

---

Alumno: Martín La Loggia

## Práctico 5 Modularización (Procedimientos)

### Alcance Archivos de Texto

#### Alcance

1. Crear un procedimiento llamado **cuadrado** que tome un número como parámetro e imprima en pantalla dicho número al cuadrado. *\*A fines pedagógicos, no a fines de elegancia de código, haremos un procedimiento en este caso y no una función como correspondería.*
  - a) Imprimir, desde dentro del procedimiento, las variables locales que este posea. Qué se imprime? *\* utilizar locals()*

```
def cuadrado(n):  
    print(n**2, locals())  
cuadrado(5)  
# salida: 25 {'n': 5}
```

- b) Agregar dos o tres variables extras con valores inventados, y volver a mostrar todas las variables locales. ¿Hubo cambio alguno?

```
def cuadrado(numero):  
    num1 = 4  
    num2 = 5  
    resultado = numero**2  
    print('1) ', resultado, num1, num2, locals())  
cuadrado(5)  
#salida: 1) 25 4 5 {'numero': 5, 'num1': 4, 'num2': 5, 'resultado': 25}
```

2. Crear una variable llamada *n*, que será global, en el código del ejercicio anterior y asignarle un valor inventado. Realizar las siguientes acciones:

a) Mostrar el valor de *n*, elevado al cuadrado desde dentro del procedimiento.

```
n = 10 # variable global
def cuadrado(n):
    print(n**2)
cuadrado(n)
# salida: 100
```

b) Modificar el valor de *n*, dentro del procedimiento. ¿Qué ocurre?

```
n = 10 # variable global
def cuadrado(n):
    n = 4
    print(n**2)
cuadrado(n)
# salida: 16
```

c) Si necesitase modificar el valor de *n*, que es una variable global, dentro del procedimiento *cuadrado*, ¿Qué debería hacer?

```
n = 10 # variable global
def cuadrado():
    global n # al definirla como globales nos referimos a la variable
             global que definimos fuera de la función, lo que nos permite
             modificar su valor
    n = 4
    print(n**2)
cuadrado()
# salida: 16
print(n)
# salida: 4
```

3. ¿Qué imprimirá en pantalla el siguiente código? ¿Cuál es el alcance de la variable frase?

El siguiente código imprimirá en pantalla el mensaje "Es un lindo día". El alcance de la variable frase es local, es decir, al definir dentro de la función una nueva variable llamada frase se imprimirá esta que es local y no la frase global que se definió al principio del código.

```
frase = "Hola"

def f():
    frase = "Es un lindo día" print(frase)
```

4. ¿Qué imprimirá en pantalla el siguiente código? Determine el alcance de cada variable.

```
x = 3

def f():
    y = x + 1 print(x)

    def g():
        x = 1
        print(y) print(x)

    g()

f()
```

```
x= 3
def f():
    y= x + 1
    print(x)
    def g():
        x = 1
        print(y)
        print(x)
    g()
f()
# salida es: 3, 4, 1

# x tiene un alcance global
# y tiene un alcance local
# g tiene un alcance local
# x dentro de la función g() tiene un alcance local
```

## Archivos de Texto

5. Crear un archivo de texto con el bloc de notas de Windows, y guardar una palabra por línea. Crear luego un programa que muestre en pantalla, una por una, las palabras que hay guardadas en ese archivo. Mostrar además, el total de líneas que había en el archivo.

```
archivo = open('archivo.txt','r')
lineas = archivo.readlines()
total = 0
for i in lineas:
    print(i)
    total += 1
print('Total de lineas: ',total)
archivo.close()
```

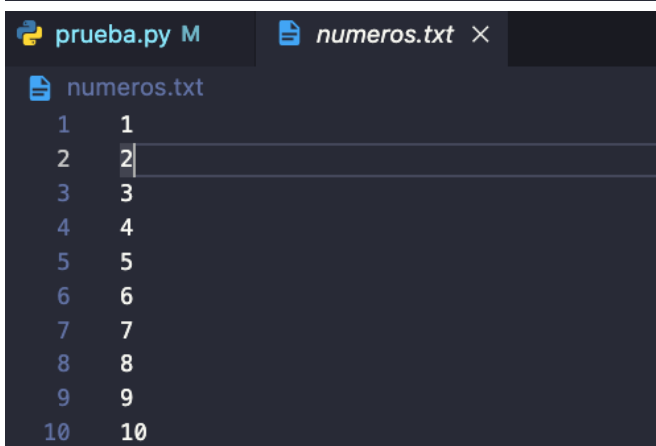
'''  
Salida:

Hola  
mundo  
esto  
es  
un  
archivo  
de  
texto


Total de lineas: 8

6. Crear un programa que genere un archivo de texto llamado *numeros.txt* con 10 números enteros guardados en el mismo, uno por línea.

```
import os
file = open("numeros.txt", "w")
text = ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10"]
for i in range(0,10):
    file.write( text[i]+ os.linesep)
file.close()
```



7. Crear un programa que le pida al usuario 5 colores, y los guarde en un archivo de texto llamado *colores.txt*

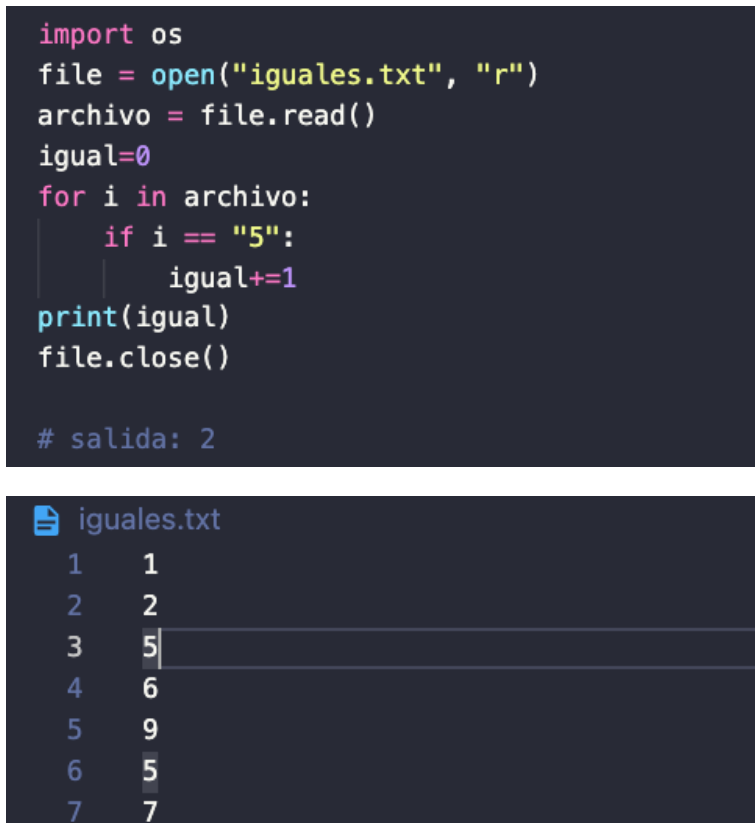


```
prueba.py M x colores.txt

prueba.py > ...
1 import os
2 file = open("colores.txt", "w")
3 for i in range(0,5):
4     color = input("Escribe un color: ")
5     file.write( color+ os.linesep)
6 file.close()
7

colores.txt
1 rojo
2 azul
3 verde
4 violeta
5 blanco
```

8. Crear un programa que dado un archivo de números con valores entre 1 y 10 (*lo puede generar como ud. desee*) determine cuantos números iguales a 5 hay en el archivo.



```
import os
file = open("iguales.txt", "r")
archivo = file.read()
igual=0
for i in archivo:
    if i == "5":
        igual+=1
print(igual)
file.close()

# salida: 2

iguales.txt
1 1
2 2
3 5
4 6
5 9
6 5
7 7
```

9. Crear un programa que dado un archivo de texto, que contiene números (*lo puede generar como ud. desee*) determine el valor promedio y la suma de todos ellos.

```
import os
file = open("numeros.txt", "r").readlines()
promedio=0
suma=0

for i in file:
    n = int(i)
    suma = suma + n
promedio = suma/len(file)

print(f'Suma: {suma}\nPromedio: {promedio}')
file.close()

# salida: Suma: 55   Promedio: 5.5
```

 numeros.txt

```
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
10   10
```

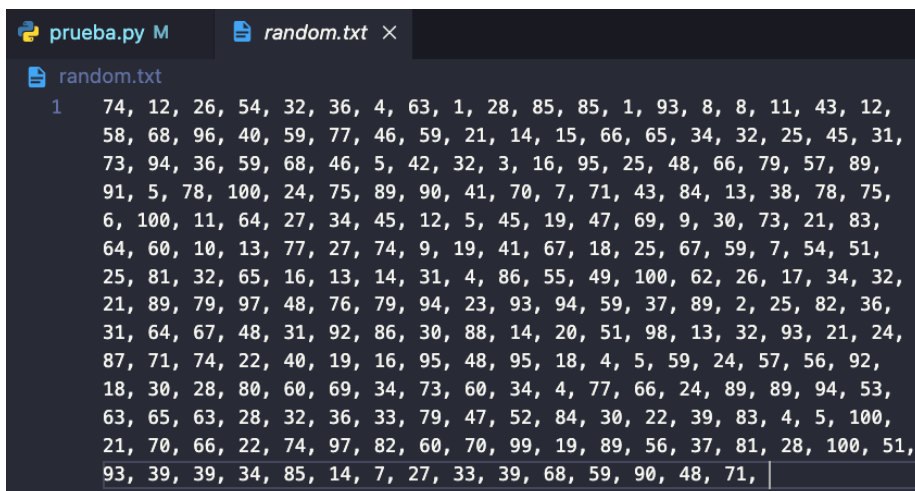
## Procedimientos

10. Crear un procedimiento que se encargue de crear un archivo de texto, con el nombre que se le de como argumento. Y que lo llene con 250 números al azar entre 1 y 100.

```
import os, random
nombre = str(input("Introduce el nombre del archivo: "))
file = open(nombre+".txt", 'w')

for i in range(0, 250):
    num = str(random.randint(0, 100))
    file.write(num+" ")

file.close()
```



```
prueba.py M random.txt x
random.txt
1 74, 12, 26, 54, 32, 36, 4, 63, 1, 28, 85, 85, 1, 93, 8, 8, 11, 43, 12,
58, 68, 96, 40, 59, 77, 46, 59, 21, 14, 15, 66, 65, 34, 32, 25, 45, 31,
73, 94, 36, 59, 68, 46, 5, 42, 32, 3, 16, 95, 25, 48, 66, 79, 57, 89,
91, 5, 78, 100, 24, 75, 89, 90, 41, 70, 7, 71, 43, 84, 13, 38, 78, 75,
6, 100, 11, 64, 27, 34, 45, 12, 5, 45, 19, 47, 69, 9, 30, 73, 21, 83,
64, 60, 10, 13, 77, 27, 74, 9, 19, 41, 67, 18, 25, 67, 59, 7, 54, 51,
25, 81, 32, 65, 16, 13, 14, 31, 4, 86, 55, 49, 100, 62, 26, 17, 34, 32,
21, 89, 79, 97, 48, 76, 79, 94, 23, 93, 94, 59, 37, 89, 2, 25, 82, 36,
31, 64, 67, 48, 31, 92, 86, 30, 88, 14, 20, 51, 98, 13, 32, 93, 21, 24,
87, 71, 74, 22, 40, 19, 16, 95, 48, 95, 18, 4, 5, 59, 24, 57, 56, 92,
18, 30, 28, 80, 60, 69, 34, 73, 60, 34, 4, 77, 66, 24, 89, 89, 94, 53,
63, 65, 63, 28, 32, 36, 33, 79, 47, 52, 84, 30, 22, 39, 83, 4, 5, 100,
21, 70, 66, 22, 74, 97, 82, 60, 70, 99, 19, 89, 56, 37, 81, 28, 100, 51,
93, 39, 39, 34, 85, 14, 7, 27, 33, 39, 68, 59, 90, 48, 71,
```

11. Crear una segunda versión del procedimiento anterior, que ahora tome dos parámetros extras,  $a$  y  $b$  para poder indicarle el intervalo de valores que se desean para los números al azar. O sea, ahora el procedimiento generará un archivo de texto, del nombre que se le de, con valores al azar en  $[a, b]$ .

```
import os, random
nombre = str(input("Introduce el nombre del archivo: "))
a = int(input("Introduce un número: "))
b = int(input("Introduce otro número: "))

file = open(nombre+".txt", 'w')

for i in range(0, 250):
    if a > b:
        num = str(random.randint(b, a))
    else:
        num = str(random.randint(a, b))

    file.write(num+" ")

file.close()
```

12. Realizar un procedimiento que tome como parámetro una longitud e imprima en pantalla un rectángulo de numerales, hueco por dentro. Por ejemplo, si se ingresó 4, se verá en pantalla: *Tip: Puede ser útil pensarlo por línea horizontal*  
Generalizarlo, luego, en una versión 2, para un parámetro extra: el carácter que se usará para dibujar el rectángulo, en vez de usar siempre un numeral.

```
####  
# #  
# #  
####
```

```
longitud = int(input("Introduce la longitud del rectangulo: "))  
estilo = str(input("Introduce el estilo del rectangulo: "))  
  
# vertical  
for i in range(0, longitud):  
    if (i == 0 or i == longitud-1):  
        print(estilo*4)  
    else:  
        print(estilo+" "*2+estilo)  
''' salida:  
####  
# #  
# #  
# #  
####  
'''  
  
# horizontal  
for i in range(0,4):  
    if (i == 0 or i == 3):  
        print(estilo*longitud)  
    else:  
        print(estilo+" "*(longitud-2)+estilo)  
''' salida:  
#####  
#      #  
#      #  
#####  
'''
```



## Variado

13. Crear un programa que permita al usuario ingresar un número en base 10 y lo devuelva en base binaria.

```
# método 1
numero = int(input("Introduce un número: ")) # entrada 9
binario = bin(numero)
print(binario) # salida: 0b1001

# método 2
numero2 = int(input("Introduce un número: ")) # entrada 9

def binario(numero):
    bit = ''
    while numero != 0:
        bit = str(numero % 2) + bit
        numero = numero // 2
    return str(numero) + bit
print(binario(numero2)) # salida: 1001
```

14. Crear un programa que permita al usuario ingresar un número en base 10 y la base a la cual desea convertirlo. Mostrar el resultado de la conversión. Siempre que la base sea menor a 10.

```
numero = int(input("Introduce un número en base 10: ")) # entrada 33
base = int(input("Introduce la base: ")) # entrada 8

def binario(numero, base):
    bit = ''
    while numero != 0:
        bit = str(numero % base) + bit
        numero = numero // base
    return str(numero) + bit
if (numero > 0):
    print(binario(numero, base)) # salida: 041
```

15. Crear un programa que permita ingresar un número en base 2, y lo convierta a base 10.

```
numero = str(input("Introduce un número en base 2: ")) # entrada 0101011
def decimal(numero): return int(numero, 2)
print(decimal(numero)) # salida: 43
```